Instituto Politécnico Nacional

Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas

Diseño Digital

Practica 0

Introducción Entorno VDHL

Equipo 11

Maldonado González César Abdiel

Ordoñez Sanchez César Leonardo

1TM8

**Practica 0: Introducción a Entorno VHDL**

Objetivo:

Que el alumno conozca la herramienta de desarrollo ISE Design 9.0 o superior para proyectos en VHDL, y aprender a simular dichos programas en HDLSim así como gravarlos en un dispositivo PLD.

Introducción Teórica:

El lenguaje de descripción hardware VHDL (Very high speed Hardware Description Logic) es un lenguaje orientado a la descripción de hardware pero con muchos elementos heredados de otros lenguajes como C o Pascal. Una vez realizado un programa en VHDL (con extensión VHD) y haberlo compilado con éxito, tendremos un fichero con el mismo nombre y extensión JED, con el cual podremos grabar una PLD (Dispositivo Lógico Programable) con la misma operatividad que el fichero VHD.

Al describir cualquier dispositivo en VHDL (desde una simple puerta and hasta un sistema completo) se deben definir dos elementos principales:

Entidad o **entity** que es la interfaz del dispositivo con el exterior. Tiene por objeto decir que señales son visibles o accesibles desde el exterior, es decir los puertos o **ports** del dispositivo.

**entity** prog01 is

**port**( a,b,c: in std\_logic;

x: out std\_logic;

**end** prog01;

Arquitectura o **arquitecture** que es la funcionalidad que realiza el dispositivo, es decir, qué transformaciones se realizarán sobre los datos que entren por los puertos de entrada para producir la salida. Dentro de este apartado es donde se dota de operatividad al circuito. Su estructura general es la siguiente, y debe estar incluida en el mismo fichero de la entidad a la que hace referencia:

**arquitecture** nombre **of** nombre\_entidad **is**

**begin**

sentencias

**end** nombre;

Para acabar esta introducción deberemos tener en cuenta una serie de detalles más de este lenguaje:

* VHDL no distingue las mayúsculas de las minúsculas, por lo que deberemos tener cuidado al asignar nombres a las variables, especialmente si estamos acostumbrados a trabajar con C.
* Las variables deben empezar por una letra, no deben contener ni espacios ni símbolos como &, %, $, #, !, etc. Su longitud no está limitada, no pueden acabar con un carácter de subrayado o tener dos subrayados seguidos.
* Para representar un número de una sola cifra, deberemos situarlo entre apóstrofes; así: '1'
* Para representar un número de más de una cifra, lo representaremos así: "10011"
* Es muy probable que en cada práctica encuentres varias entidades y varias arquitecturas. Tomando como ejemplo al multiplexor, sabemos que no todos tienen el mismo número de bits o de canales, por eso cada uno tiene una entidad distinta.

Operadores lógicos que acepta VHDL

and nand

or nor

xor xnor

not

Procedimiento:

1.- Realizar la reducción de la siguiente expresión booleana con las propiedades del algebra de Boole, dibujar la solución con simbología y una vez que tengas la solución comprobar con tablas la verdad y también comprobar con un PLD las ecuaciones resultantes.

Ej. Solución

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | A´+B+C | COMPROBACIÓN |
| 0 | 0 | 0 | 1 |  |
| 0 | 0 | 1 | 1 |  |
| 0 | 1 | 0 | 1 |  |
| 0 | 1 | 1 | 1 |  |
| 1 | 0 | 0 | 0 |  |
| 1 | 0 | 1 | 1 |  |
| 1 | 1 | 0 | 1 |  |
| 1 | 1 | 1 | 1 |  |

Texto

Descripción generada automáticamente

Diagrama, Texto

Descripción generada automáticamente2.- Al Igual que en el punto anterior, hacer el mismo procedimiento para las operaciones siguientes.

**Conversión de Variables**

**X = A , Y = B, Z=C, W=D**

|  |
| --- |
| **1** |



**Transformación de la expresión.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | A´+B´+C´ | COMPROBACIÓN |
| 0 | 0 | 0 | 1 |  |
| 0 | 0 | 1 | 1 |  |
| 0 | 1 | 0 | 1 |  |
| 0 | 1 | 1 | 1 |  |
| 1 | 0 | 0 | 1 |  |
| 1 | 0 | 1 | 1 |  |
| 1 | 1 | 0 | 1 |  |
| 1 | 1 | 1 | 0 |  |

|  |
| --- |
| **Programa** |

|  |
| --- |
| **2** |

****

**Transformación de la expresión.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | D | B+D | COMPROBACIÓN |
| 0 | 0 | 0 | 0 | 0 |  |
| 0 | 0 | 0 | 1 | 1 |  |
| 0 | 0 | 1 | 0 | 0 |  |
| 0 | 0 | 1 | 1 | 1 |  |
| 0 | 1 | 0 | 0 | 1 |  |
| 0 | 1 | 0 | 1 | 1 |  |
| 0 | 1 | 1 | 0 | 1 |  |
| 0 | 1 | 1 | 1 | 1 |  |
| 1 | 0 | 0 | 0 | 0 |  |
| 1 | 0 | 0 | 1 | 1 |  |
| 1 | 0 | 1 | 0 | 0 |  |
| 1 | 0 | 1 | 1 | 1 |  |
| 1 | 1 | 0 | 0 | 1 |  |
| 1 | 1 | 0 | 1 | 1 |  |
| 1 | 1 | 1 | 0 | 1 |  |
| 1 | 1 | 1 | 1 | 1 |  |

|  |
| --- |
| **Programa** |

|  |
| --- |
| **3** |

****

**Transformación de la expresión.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | D | (A+B)(A´+B´)(A+C´+D´) | COMPROBACIÓN |
| 0 | 0 | 0 | 0 | 0 |  |
| 0 | 0 | 0 | 1 | 0 |  |
| 0 | 0 | 1 | 0 | 0 |  |
| 0 | 0 | 1 | 1 | 0 |  |
| 0 | 1 | 0 | 0 | 1 |  |
| 0 | 1 | 0 | 1 | 1 |  |
| 0 | 1 | 1 | 0 | 1 |  |
| 0 | 1 | 1 | 1 | 1 |  |
| 1 | 0 | 0 | 0 | 0 |  |
| 1 | 0 | 0 | 1 | 1 |  |
| 1 | 0 | 1 | 0 | 1 |  |
| 1 | 0 | 1 | 1 | 1 |  |
| 1 | 1 | 0 | 0 | 0 |  |
| 1 | 1 | 0 | 1 | 0 |  |
| 1 | 1 | 1 | 0 | 0 |  |
| 1 | 1 | 1 | 1 | 0 |  |

|  |
| --- |
| **Programa** |

3.- Diseñar en el lenguaje de descripción de hardware, los siguientes integrados, 74LS00, 74L04, 74L86, 74L10, 74LS21 y 74ls30.

**74LS00:** La 74LS00 Consta de 4 compuertas NAND-positivas de 2 entradas, con estas compuertas lógicas podemos invertir una señal o generar una lógica para activar otro circuito integrado.

Este circuito y sus variantes es uno de los utilizados en la electrónica, cuando tienes que implementar funciones que usen muchas compuertas NAND.

Un dibujo de una persona

Descripción generada automáticamente con confianza baja

H

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | OUT | SIMULACION |
| 0 | 0 | 1 |  |
| 0 | 1 | 1 |  |
| 1 | 0 | 1 |  |
| 1 | 1 | 0 |  |

**74L04:** Este circuito integrado cuenta con 6 inversores independientes, cada inversor puede ser usado sin la necesidad de conectar los demás.

Su salida es el estado inverso a su entrada, la cual no puede ser superior al voltaje de alimentación del circuito integrado.

Diagrama

Descripción generada automáticamente con confianza media

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | OUT | SIMULACION |
| 0 | 1 | 1 |  |
| 1 | 0 | 0 |  |

**74L86:** Es una compuerta OR-exclusiva de 2 entradas cuádruple y cuatro compuertas independientes de XOR de 2 entradas. Realiza funciones booleanas en lógica positiva.

Diagrama

Descripción generada automáticamenteUna aplicación común es como un elemento completo/verdadero. Si una de las entradas es baja, la otra se reproducirá en forma real en salida.

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | OUT | SIMULACION |
| 0 | 0 | 0 |  |
| 0 | 1 | 1 |  |
| 1 | 0 | 1 |  |
| 1 | 1 | 0 |  |

**74L10:** Este circuito integrado consta de tres compuertas NAND de tres entradas independientes, se caracteriza por la operación catálogo.

Diagrama

Descripción generada automáticamente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | Z | OUT | SIMULACION |
| 0 | 0 | 0 | 1 |  |
| 0 | 0 | 1 | 1 |  |
| 0 | 1 | 0 | 1 |  |
| 0 | 1 | 1 | 1 |  |
| 1 | 0 | 0 | 1 |  |
| 1 | 0 | 1 | 1 |  |
| 1 | 1 | 0 | 1 |  |
| 1 | 1 | 1 | 0 |  |

**74LS21**: Este es una compuerta doble AND positiva de 4 entradas y dos Compuertas AND de 4 entradas independientes.

Diagrama

Descripción generada automáticamente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | Z | W | OUT |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Diagrama

Descripción generada automáticamente**74LS30**: Circuito integrado diseñado como compuerta NAND de ocho entradas, sus aplicaciones son en la comunicación y en redes.

|  |  |
| --- | --- |
| ENTRADA | SALIDA |
| CUALQUIER ENTRADA 0 | 1 |
| TODAS LAS ENTRADAS 1 | 0 |

**CONCLUSIONES.**

**Maldonado González César Abdiel**

**Ordoñez Sánchez César Leonardo**

En esta practica aprendimos la importancia tan grande que tiene la algebra de Boole, como nos puede ayudar en reducir un circuito sin afectar sus valores, ayudándonos a hacer un circuito más sencillo de entender, programar y recrear, así mismo nos introdujo a el desarrollo de los programas en el ISE para acostumbrarnos a este.